

CT/PTD 4 MAY 2001

IMAGE DESCRIPTION SYSTEM AND METHOD

## SPECIFICATION

CROSS REFERENCE TO RELATED APPLICATIONS

5 This application claims priority to United States provisional patent application Serial No. 60/118,020, filed February 1, 1999, United States provisional patent application serial no. 60/118,027, filed February 1, 1999 and United States provisional patent application serial no. 60/107,463, filed November 6, 1998.

FIELD OF THE INVENTION

10 The present invention relates to techniques for describing multimedia information, and more specifically, to techniques which describe image information and the content of such information.

BACKGROUND OF THE INVENTION

15 With the maturation of the global Internet and the widespread employment of regional networks and local networks, digital multimedia information has become increasingly accessible to consumers and businesses. Accordingly, it has become progressively more important to develop systems that process, filter, search and organize digital multimedia information, so that useful information can be culled from this growing mass of raw information.

20 At the time of filing the instant application, solutions exist that allow consumers and business to search for textual information. Indeed, numerous text-based search engines, such as those provided by yahoo.com, goto.com, excite.com and others are available on the World Wide Web, and are among the most visited Web sites, indicating the significant of the demand for such information retrieval technology.

25 Unfortunately, the same is not true for multimedia content, as no generally recognized description of this material exists. In this regard, there have

been past attempts to provide multimedia databases which permit users to search for pictures using characteristics such as color, texture and shape information of video objects embedded in the picture. However, at the closing of the 20th Century, it is not yet possible to perform a general search the Internet or most regional or local  
5 networks for multimedia content, as no broadly recognized description of this material exists. Moreover, the need to search for multimedia content is not limited to databases, but extends to other applications, such as digital broadcast television and multimedia telephony.

One industry wide attempt to develop such a standard multimedia  
10 description framework has been through the Motion Pictures Expert Group's ("MPEG") MPEG-7 standardization effort. Launched in October 1996, MPEG-7 aims to standardize content descriptions of multimedia data in order to facilitate content-focused applications like multimedia searching, filtering, browsing and summarization. A more complete description of the objectives of the MPEG-7  
15 standard are contained in the International Organisation for Standardisation document ISO/IEC JTC1/SC29/WG11 N2460 (Oct. 1998), the content of which is incorporated by reference herein.

The MPEG-7 standard has the objective of specifying a standard set of descriptors as well as structures (referred to as "description schemes") for the  
20 descriptors and their relationships to describe various types of multimedia information. MPEG-7 also proposes to standardize ways to define other descriptors as well as "description schemes" for the descriptors and their relationships. This description, i.e. the combination of descriptors and description schemes, shall be associated with the content itself, to allow fast and efficient searching and filtering for  
25 material of a user's interest. MPEG-7 also proposes to standardize a language to specify description schemes, i.e. a Description Definition Language ("DDL"), and the schemes for binary encoding the descriptions of multimedia content.

At the time of filing the instant application, MPEG is soliciting proposals for techniques which will optimally implement the necessary description  
30 schemes for future integration into the MPEG-7 standard. In order to provide such optimized description schemes, three different multimedia-application arrangements

can be considered. These are the distributed processing scenario, the content-exchange scenario, and the format which permits the personalized viewing of multimedia content.

Regarding distributed processing, a description scheme must provide  
5 the ability to interchange descriptions of multimedia material independently of any platform, any vendor, and any application, which will enable the distributed processing of multimedia content. The standardization of interoperable content descriptions will mean that data from a variety of sources can be plugged into a variety of distributed applications, such as multimedia processors, editors, retrieval  
10 systems, filtering agents, etc. Some of these applications may be provided by third parties, generating a sub-industry of providers of multimedia tools that can work with the standardized descriptions of the multimedia data.

A user should be permitted to access various content providers' web sites to download content and associated indexing data, obtained by some low-level or  
15 high-level processing, and proceed to access several tool providers' web sites to download tools (e.g. Java applets) to manipulate the heterogeneous data descriptions in particular ways, according to the user's personal interests. An example of such a multimedia tool will be a video editor. A MPEG-7 compliant video editor will be able to manipulate and process video content from a variety of sources if the description  
20 associated with each video is MPEG-7 compliant. Each video may come with varying degrees of description detail, such as camera motion, scene cuts, annotations, and object segmentations.

A second scenario that will greatly benefit from an interoperable content-description standard is the exchange of multimedia content among  
25 heterogeneous multimedia databases. MPEG-7 aims to provide the means to express, exchange, translate, and reuse existing descriptions of multimedia material.

Currently, TV broadcasters, Radio broadcasters, and other content providers manage and store an enormous amount of multimedia material. This material is currently described manually using textual information and proprietary  
30 databases. Without an interoperable content description, content users need to invest manpower to translate manually the descriptions used by each broadcaster into their

own proprietary scheme. Interchange of multimedia content descriptions would be possible if all the content providers embraced the same content description schemes. This is one of the objectives of MPEG-7.

5 Finally, multimedia players and viewers that employ the description schemes must provide the users with innovative capabilities such as multiple views of the data configured by the user. The user should be able to change the display's configuration without requiring the data to be downloaded again in a different format from the content broadcaster.

The foregoing examples only hint at the possible uses for richly  
10 structured data delivered in a standardized way based on MPEG-7. Unfortunately, no prior art techniques available at present are able to generically satisfy the distributed processing, content-exchange, or personalized viewing scenarios. Specifically, the prior art fails to provide a technique for capturing content embedded in multimedia information based on either generic characteristics or semantic relationships, or to  
15 provide a technique for organizing such content. Accordingly, there exists a need in the art for efficient content description schemes for generic multimedia information.

### SUMMARY OF THE INVENTION

It is an object of the present invention to provide a description system for image content.

20 It is a further object of the present invention to provide a description system for image content which is extensible.

It is another object of the present invention to provide a description system for image content which is scalable.

25 It is yet another object of the present invention to provide a description system for image content which satisfies the requirements of certain proposed media standards, such as the MPEG-7 standard.

It is an object of the present invention to provide a method for describing image content.

30 It is a further object of the present invention to provide a method for describing image content which is extensible.

It is another object of the present invention to provide a method for describing image content which is scalable.

It is yet another object of the present invention to provide a method for describing image content which satisfies the requirements of certain proposed media standards, such as the MPEG-7 standard.

### BRIEF DESCRIPTION OF THE DRAWING

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

10        Figure 1A is an exemplary image for the image description system of the present invention.

Figure 1B is an exemplary object hierarchy for the image description system of the present invention.

15        Figure 1C is an exemplary entity relation graph for the image description system of the present invention.

Figure 2 is an exemplary block diagram of the image description system of the present invention.

Figure 3A is an exemplary object hierarchy for the image description system of the present invention.

20        Figure 3B is another exemplary object hierarchy for the image description system of the present invention.

Figure 4A is a representation of an exemplary image for the image description system of the present invention.

25        Figure 4B is an exemplary clustering hierarchy for the image description system of the present invention.

Figure 5 is an exemplary block diagram of the image description system of the present invention.

Figure 6 is an exemplary process flow diagram for the image description system of the present invention.

Figure 7 is an exemplary block diagram of the image description system of the present invention.

Figure 8 is an another exemplary block diagram of the image description system of the present invention.

5 Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can  
10 be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

#### DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention constitutes a description scheme (DS) for images, wherein simple but powerful structures representing generic image data are  
15 utilized. Although the description scheme of the present invention can be used with any type of standard which describes image content, a preferred embodiment of the invention is used with the MPEG-7 standard. Although any Description Definition Language (DDL) may be used to implement the DS of the present invention, a preferred embodiment utilizes the eXtensible Markup Language (XML), which is a  
20 streamlined subset of SGML (Standard Generalized Markup Language, ISO 8879) developed specifically for World Wide Web applications. SGML allows documents to be self-describing, in the sense that they describe their own grammar by specifying the tag set used in the document and the structural relationships that those tags represent. XML retains the key SGML advantages in a language that is designed to  
25 be vastly easier to learn, use, and implement than full SGML. A complete description of XML can be found at the World Wide Web Consortium's web page on XML, at <http://www.w3.org/XML/>, the contents of which is incorporated by reference herein.

The primary components of a characterization of an image using the  
30 description scheme of the present invention are objects, feature classifications, object

hierarchies, entity-relation graphs, multiple levels of abstraction, code downloading, and modality transcoding, all of which will be described in additional detail below. In the description scheme of the present invention, an image document is represented by a set of objects and relationships among objects. Each object may have one or more associated features, which are generally grouped into the following categories: media features, visual features, and semantic features. Each feature can include descriptors that can facilitate code downloading by pointing to external extraction and similarity matching code. Relationships among objects can be described by object hierarchies and entity-relation graphs. Object hierarchies can also include the concept of multiple levels of abstraction. Modality transcoding allows user terminals having different capabilities (such as palmpilots, cellular telephones, or different types of personal computers (PC's), for example) to receive the same image content in different resolutions and/or different modalities.

As described above, a preferred embodiment of the image description system of the present invention is used with the MPEG-7 standard. In accord with this standard, this preferred embodiment uses objects as the fundamental entity in describing various levels of image content, which can be defined along different dimensions. For example, objects can be used to describe image regions or groups of image regions. High-level objects can in turn be used to describe groups of primitive objects based on semantics or visual features. In addition, different types of features can be used in connection with different levels of objects. For instance, visual features can be applied to objects corresponding to physical components in the image content, whereas semantic features can be applied to any level of object.

In addition, the image description system of the present invention provides flexibility, extensibility, scalability and convenience of use. In the interest of enhanced flexibility, the present invention allows portions of the image description system to be instantiated, uses efficient categorization of features and clustering of objects by way of a clustering hierarchy, and also supports efficient linking, embedding and downloading of external feature descriptors and execution code. The present invention also provides extensibility by permitting elements defined in the description scheme to be used to derive new elements for different domains.

Scalability is provided by the present invention's capability to define multiple abstraction levels based on any arbitrary set of criteria using object hierarchies. These criteria can be specified in terms of visual features (size and color, for example), semantic relevance (relevance to user interest profile, for example) and/or service quality (media features, for example). The present invention is convenient to use because it specifies a minimal set of components: namely, objects, feature classes, object hierarchies, and entity-relation graphs. Additional objects and features can be added in a modular and flexible way. In addition, different types of object hierarchies and entity-relation graphs can each be defined in a similar fashion.

Under the image description system of the present invention, an image is represented as a set of image objects, which are related to one another by object hierarchies and entity-relation graphs. These objects can have multiple features which can be linked to external extraction and similarity matching code. These features are categorized into media, visual, and semantic features, for example. Image objects can be organized in multiple different object hierarchies. Non-hierarchical relationships among two or more objects can be described using one or more different entity-relation graphs. For objects contained in large images, multiple levels of abstraction in clustering and viewing such objects can be implemented using object hierarchies. These multiple levels of abstraction in clustering and viewing such images can be based on media, visual, and/or semantic features, for example. One example of a media feature includes modality transcoding, which permits users having different terminal specifications to access the same image content in satisfactory modalities and resolutions.

The characteristics and operation of the image description system of the present invention will now be presented in additional detail. Figs. 1A, 1B and 1C depict an exemplary description of an exemplary image in accordance with the image description system of the present invention. Fig. 1A depicts an exemplary set of image objects and exemplary corresponding object features for those objects. More specifically, Fig. 1A depicts image object 1 (i.e., O1) 2 ("Person A"), O2 6 ("Person B") and O3 4 ("People") contained in O0 8 (i.e., the overall exemplary photograph), as well as exemplary features 10 for the exemplary photograph depicted. Fig. 1B



depicts an exemplary spatial object hierarchy for the image objects depicted in Fig. 1A, wherein O0 8 (the overall photograph) is shown to contain O1 2 ("Person A") and O2 6 ("Person B"). Fig. 1C depicts an exemplary entity-relation (E-R) graph for the image objects depicted in Fig. 1A, wherein O1 2 ("Person A") is characterized as being located to the left of, and shaking hands with, O2 6 ("Person B").

Fig. 2 depicts an exemplary graphical representation of the image description system of the present invention, utilizing the conventional Unified Modeling Language (UML) format and notation. Specifically, the diamond-shaped symbols depicted in Fig. 2 represent the composition relationship. The range associated with each element represents the frequency in that composition relationship. Specifically, the nomenclature "0...\*" denotes "greater than or equal to 0;" the nomenclature "1...\*" denotes "greater than or equal to 1."

In the following discussion, the text appearing between the characters "<" and ">" denotes the characterization of the referenced elements in the XML preferred embodiments which appear below. In the image description system of the present invention as depicted in Fig. 2, an image element 22 (<image>), which represents an image description, includes an image object set element 24 (<image\_object\_set>), and may also include one or more object hierarchy elements 26 (<object\_hierarchy>) and one or more entity-relation graphs 28 (<entity\_relation\_graph>). Each image object set element 24 includes one or more image object elements 30. Each image object element 30 may include one or more features, such as media feature elements 36, visual feature elements 38 and/or semantic feature elements 40. Each object hierarchy element 26 contains an object node element 32, each of which may in turn contain one or more additional object node elements 32. Each entity-relation graph 28 contains one or more entity relation elements 34. Each entity relation element 34 in turn contains a relation element 44, and may also contain one or more entity node elements 42.

An object hierarchy element 26 is a special case of an entity-relation graph 28, wherein the entities are related by containment relationships. The preferred embodiment of the image description system of the present invention includes object hierarchy elements 26 in addition to entity relationship graphs 28, because an object

hierarchy element 26 is a more efficient structure for retrieval than is an entity relationship graph 28. In addition, an object hierarchy element 26 is the most natural way of defining composite objects, and MPEG-4 objects are constructed using hierarchical structures.

5 To maximize flexibility and generality, the image description system of the present invention separates the definition of the objects from the structures that describe relationships among the objects. Thus, the same object may appear in different object hierarchies 26 and entity-relation graphs 28. This avoids the undesirable duplication of features for objects that appear in more than one object  
10 hierarchy 26 and/or entity-relation graph 28. In addition, an object can be defined without the need for it to be included in any relational structure, such as an object hierarchy 26 or entity-relation graph 28, so that the extraction of objects and relations among objects can be performed at different stages, thereby permitting distributed processing of the image content.

15 Referring to Figs. 1A, 1B, 1C and Fig. 2, an image object 30 refers to one or more arbitrary regions of an image, and therefore can be either continuous or discontinuous in space. In Figs. 1A, 1B and 1C, O1 2 ("Person A"), O2 6 ("Person B"), and O0 8 (i.e., the photograph) are objects with only one associated continuous region. On the other hand, O3 4 ("People") is an example of an object composed of  
20 multiple regions separated from one another in space. A global object contains features that are common to an entire image, whereas a local object contains only features of a particular section of that image. Thus, in Figs. 1A, 1B and 1C, O0 8 is a global object representing the entire image depicted, whereas O1 2, O2 4 and O3 4 are each local objects representing a person or persons contained within the overall image.

25 Various types of objects which can be used in connection with the present invention include visual objects, which are objects defined by visual features such as color or texture; media objects; semantic objects; and objects defined by a combination of semantic, visual, and media features. Thus, an object's type is determined by the features used to describe that object. As a result, new types of  
30 objects can be added as necessary. In addition, different types of objects may be

derived from these generic objects by utilizing inheritance relationships, which are supported by the MPEG-7 standard.

As depicted in Fig. 2, the set of all image object elements 30 (<image\_object>) described in an image is contained within the image object set element 24 (<image\_object\_set>). Each image object element 30 can have a unique identifier within an image description. The identifier and the object type (e.g., local or global) are expressed as attributes of the object element ID and type, respectively. An exemplary implementation of an exemplary set of objects to describe the image depicted in Figs. 1A, 1B and 1C is shown below listed in XML. In all XML listings shown below, the text appearing between the characters "<!--" and "-->" denotes comments to the XML code:

```
<image_object_set>
  <image_object id="O0" type="GLOBAL"> </image_object> <!-- Photograph -->
  <image_object id="O1" type="LOCAL"> </image_object> <!-- Person A -->
  <image_object id="O2" type="LOCAL"> </image_object> <!-- Person B -->
  <image_object id="O3" type="LOCAL"> </image_object> <!-- People -->
</image_object_set>
```

As depicted in Fig. 2, image objects 30 may for example contain three feature class elements that group features together according to the information conveyed by those features. Examples of such feature class elements include media features 36 (<img\_obj\_media\_features>), visual features 38 (<img\_obj\_visual\_features>), and semantic features 40 (<img\_obj\_semantic\_features>). Table 1 below denotes an exemplary list of features for each of these feature classes.

**Table 1: Exemplary Feature Classes and Features.****Feature Class Features**

Semantic      Text Annotation, Who, What Object, What Action, Why, When,  
Where

5      Visual Color, Texture, Position, Size, Shape, Orientation

Media File Format, File Size, Color Representation, Resolution, Data File Location,  
Modality Transcoding, Author, Date of Creation

Each feature element contained in the feature classes in an image  
object element 30 will include descriptors in accordance with the MPEG-7 standard.

10      Table 2 below denotes exemplary descriptors that may be associated with certain of  
the exemplary visual features denoted in Table 1. Specific descriptors such as those  
denoted in Table 2 may also contain links to external extraction and similarity  
matching code. Although Tables 1 and 2 denote exemplary features and descriptors,  
the image description system of the present invention may include, in an extensible  
15      and modular fashion, any number of features and descriptors for each object.

**Table 2: Exemplary Visual Features and Associated Descriptors.**

<b>Feature</b>	<b>Descriptors</b>
Color	Color Histogram, Dominant Color, Color Coherence Vector, Visual Sprite Color
5 Texture	Tamura, MSAR, Edge Direction Histogram, DCT Coefficient Energies, Visual Sprite Texture
Shape	Bounding Box, Binary Mask, Chroma Key, Polygon Shape, Fourier Shape, Boundary, Size, Symmetry, Orientation

The XML example shown below denotes an example of how features and descriptors can be defined to be included in an image object 30. In particular, the below example defines the exemplary features 10 associated with the global object O0 depicted in Figs. 1A, 1B and 1C, namely, two semantic features (“where” and “when”), one media feature (“file format”), and one visual feature (“color” with a “color histogram” descriptor). An object can be described by different concepts (<concept>) in each of the semantic categories as shown in the example below.

```

<image_object id="O0" type="GLOBAL"> <!-- Global object: Photograph -->
  <img_obj_semantic_features>
    <where>
      <concept> Columbia University, NYC </concept>
      <concept> Outdoors </concept>
    </where>
    <when> <concept> 5/31/99 </concept> </when>
  </img_obj_semantic_features>
  <img_obj_media_features>
    <file_format> JPG </file_format>
  </img_obj_media_features>
  <img_obj_visual_features>
    <color>

```

```

        <color_histogram>
            <value format="float[166]"> .3 .03 .45 ... </value>
        </color_histogram>
    </color>
5    </img_obj_visual_features>
    </image_global_object>

```

As depicted in Fig. 2, in the image description system of the present invention the object hierarchy element 26 can be used to organize the image objects 30 in the image object set 24, based on different criteria such as media features 36, visual features 38, semantic features 40, or any combinations thereof. Each object hierarchy element 26 constitutes a tree of object nodes 32 which reference image object elements 30 in the image object set 24 via link 33.

An object hierarchy 26 involves a containment relation from one or more child nodes to a parent node. This containment relation may be of numerous different types, depending on the particular object features being utilized, such as media features 36, visual features 38 and/or semantic features 40, for example. For example, the spatial object hierarchy depicted in Fig. 1B describes a visual containment, because it is created in connection with a visual feature, namely spatial position. Figs. 3A and 3B depict two additional exemplary object hierarchies.

Specifically, Fig. 3A depicts an exemplary hierarchy for the image objects depicted in Fig. 1A, based on the "who" semantic feature as denoted in Table 1. Thus, in Fig. 3A, O3 4 ("People") is shown to contain O1 2 ("Person A") and O2 6 ("Person B"). Fig. 3B depicts an exemplary hierarchy based on exemplary color and shape visual features such as those denoted in Table 1. In Fig. 3B, O7 46 could for example be defined to be the corresponding region of an object satisfying certain specified color and shape constraints. Thus, Fig. 3B depicts O7 46 ("Skin Tone & Shape") as containing O4 48 ("Face Region 1") and O6 50 ("Face Region 2"). Object hierarchies 26 combining different features can also be constructed to satisfy the requirements of a broad range of application systems.

As further depicted in Fig. 2, each object hierarchy element 26 (<object\_hierarchy>) contains a tree of object nodes (ONs) 32. The object hierarchies

also may include optional string attribute types. If such string attribute types are present, a thesaurus can provide the values of these string attribute types so that applications can determine the types of hierarchies which exist. Every object node 32 (<object\_node>) references an image object 30 in the image object set 24 via link 33.

5 Image objects 30 also can reference back to the object nodes 32 referencing them via link 33. This bi-directional linking mechanism permits efficient transversal from image objects 30 in the image object set 24 to the corresponding object nodes 32 in the object hierarchy 26, and vice versa. Each object node 32 references an image object 30 through an attribute (object\_ref) by using a unique identifier of the image  
10 object. Each object node 32 may also contain a unique identifier in the form of an attribute. These unique identifiers for the object nodes 32 enable the objects 30 to reference back to the object nodes which reference them using another attribute (object\_node\_ref). An exemplary XML implementation of the exemplary spatial object hierarchy depicted in Fig. 1B is expressed below.

```
15      <object_hierarchy type="SPATIAL"> <!-- Object hierarchy: spatial hierarchy -->
          <object_node id="ON0" object_ref="O0"> <!-- Photograph -->
              <object_node id="ON1" object_ref="O1"> </object_node> <!-- Person A -->
              <object_node id="ON2" object_ref="O2"> </object_node> <!-- Person B -->
          </object_node>
20      </object_hierarchy>
```

Object hierarchies 26 can also be used to build clustering hierarchies and to generate multiple levels of abstraction. In describing relatively large images, such as satellite photograph images for example, a problem normally arises in describing and retrieving, in an efficient and scalable manner, the many objects  
25 normally contained in such images. Clustering hierarchies can be used in connection with the image description system of the present invention to provide a solution to this problem.

Figs. 4A and 4B depict an exemplary use of a clustering hierarchy scheme wherein objects are clustered hierarchically based on their respective size (<size>). In particular, Fig. 4A depicts a representation of a relatively large image,  
30

such as a satellite photograph image for example, wherein objects O11 52, O12 54, O13 56, O14 58 and O15 60 represent image objects of varying size, such as lakes on the earth's surface for example, contained in the large image. Fig. 4B represents an exemplary size-based clustering hierarchy for the objects depicted in Fig. 4A, wherein objects O11 52, O12 54, O13 56, O14 58 and O15 60 represent the objects depicted in Fig. 4A, and wherein additional objects O16 62, O17 64 and O18 56 represent objects which specify the size-based criteria for the cluster hierarchy depicted in Fig. 4B. In particular, objects O16 62, O17 64 and O18 56 may for example represent intermediate nodes 32 of an object hierarchy 26, which intermediate nodes are represented as image objects 30. These objects include the criteria, conditions and constraints related to the size feature used for grouping the objects together in the depicted cluster hierarchy. In the particular example depicted in Fig. 4B, objects O16 62, O17 64 and O18 56 are used to form a clustering hierarchy having three hierarchical levels based on size. Object O16 62 represents the size criteria which forms the clustering hierarchy. Object O17 64 represents a second level of size criteria of less than 50 units, wherein such units may represent pixels for example; object O18 56 represents a third level of size criteria of less than 10 units. Thus, as depicted in Fig. 4B, objects O11 52, O12 54, O13 56, O14 58 and O15 60 are each characterized as having a specified size of a certain number of units. Similarly, objects O13 56, O14 58 and O15 60 are each characterized as having a specified size of less than 50 units, and object O15 60 is characterized as having a specified size of less than 10 units.

Although Figs. 4A and 4B depict an example of a single clustering hierarchy based on only a single set of criteria, namely size, multiple clustering hierarchies using different criteria involving multiple features may also be used for any image. For example, such clustering hierarchies may group together objects based on any combination of media, visual, and/or semantic features. This procedure is similar to the procedure used to cluster images together in visual information retrieval engines. Each object contained within the overall large image is assigned an image object 30 in the object set 24, and may also be assigned certain associated features such as media features 36, visual features 38 or semantic features 40. The



intermediate nodes 32 of the object hierarchy 26 are represented as image objects 30, and also include the criteria, conditions and constraints related to one or more features used for grouping the objects together at that particular level. An image description may include any number of clustering hierarchies. The exemplary clustering hierarchy depicted in Figs. 4A and 4B is expressed in an exemplary XML implementation below.

```

10      <image>
      <image_object_set>
      <image_object type="LOCAL" id="O11"> <!-- Real objects of the image -->
      <size> <num_pixels> 120 </num_pixels> </size>
      </image_object> <!-- Others objects -->
      <image_object type="LOCAL" id="O17"> <!-- Intermediate nodes in the
      hierarchy-->
      <size> <num_pixels> <less_than> 50 </less_than> </num_pixels>
15  </size>
      </image_object> <!-- Others objects -->
      </image_object_set>
      <object_hierarchy>
      <object_node id="ON11" object_ref="O16">
20      <object_node id="ON12" object_ref="O11" />
      <object_node id="ON13" object_ref="O12" />
      <object_node id="ON14" object_ref="O17">
      <object_node id="ON15" object_ref="O13" />
      <object_node id="ON16" object_ref="O14" />
25      <object_node id="ON17" object_ref="O18">
      <object_node id="ON18" object_ref="O15" />
      </object_node>
      </object_node>
      </object_node>
30      </object_hierarchy>
    </image>

```

As depicted in the multiple clustering hierarchy example of Figs. 4A and 4B, and as denoted in Table 3 below, there are defined three levels of abstraction based on the size of the objects depicted. This multi-level abstraction scheme

provides a scalable method for retrieving and viewing objects in the image depicted in Fig. 4A. Such an approach can also be used to represent multiple abstraction levels based on other features, such as various semantic classes for example.

**Table 3: Objects in Each Abstraction Level**

5 <b>Abstraction Level</b>	<b>Objects</b>
1	O11, O12
2	O11, O12, O13, O14
3	O11, O12, O13, O14, O15

Additional XML embodiments of the exemplary clustering hierarchies and multiple  
10 levels of abstraction depicted in Figs. 4A and 4B are included in Appendix A.

Although such hierarchical structures are suitable for purposes of  
retrieving images, certain relationships among objects cannot adequately be expressed  
using such structures. Thus, as depicted in Figs. 1C and 2, the image description  
system of the present invention also utilizes entity-relation (E-R) graphs 28 for the  
15 specification of more complex relationships among objects. An entity-relation graph  
28 is a graph of one or more entity nodes 42 and the relationships among them. Table  
4 below denotes several different exemplary types of such relationships, as well as  
specific examples of each.

**Table 4: Examples of relation types and relations.**

<b>Relation Type</b>	<b>Relations</b>
<b>Spatial</b>	
5                      Directional	Top Of, Bottom Of, Right Of, Left Of, Upper Left Of, Upper Right Of, Lower Left Of, Lower Right Of
Topological	Adjacent To, Neighboring To, Nearby, Within, Contain
<b>Semantic</b>	Relative Of, Belongs To, Part Of, Related To, Same As, Is A, Consist Of

Entity-relation graphs can be of any general structure, and can also be  
 10 customized for any particular application by utilizing various inheritance  
 relationships. The exemplary entity-relation graph depicted in Fig. 1C describes an  
 exemplary spatial relationship, namely "Left Of", and an exemplary semantic  
 relationship, namely "Shaking Hands With", between objects O1 2 and O2 6 depicted  
 in Fig. 1A.

15                      As depicted in Fig. 2, the image description system of the present  
 invention allows for the specification of zero or more entity-relation graphs 28  
 (<entity\_relation\_graph>). An entity-relation graph 28 includes one or more sets of  
 entity-relation elements 34 (<entity\_relation>), and also contains two optional  
 attributes, namely a unique identifier ID and a string type to describe the binding  
 20 expressed by the entity relation graph 28. Values for such types could for example be  
 provided by a thesaurus. Each entity relation element 34 contains one relation  
 element 44 (<relation>), and may also contain one or more entity node elements 42  
 (<entity\_node>) and one or more entity-relation elements 34. The relation element 44  
 contains the specific relationship being described. Each entity node element 42

references an image object 30 in the image object set 24 via link 43, by utilizing an attribute, namely object\_ref. Via link 43, image objects 30 also can reference back to the entity nodes 42 referencing the image objects 30 by utilizing an attribute (entity\_code\_ref).

5 As depicted in the exemplary entity-relation graph 28 of Fig. 1C, the entity-relation graph 28 contains two entity relations 34 between object O1 2 ("Person A") and object O2 6 ("Person B"). The first such entity relation 34 describes the spatial relation 44 regarding how object O1 2 is positioned with respect to (i.e., to the "Left Of") object O2 6. The second such entity relation 34 depicted in Fig. 1C  
10 describes the semantic relation 44 of how object O1 2 is "Shaking Hand With" object O2 6. An exemplary XML implementation of the entity-relation graph example depicted in Fig. 1C is shown below:

```

15 <entity_relation_graph>
    <entity_relation> <!-- Spatial, directional entity relation -->
        <relation type="SPATIAL.DIRECTIONAL"> Left Of </relation>
        <entity_node id="ETN1" object_ref="O1"/> <entity_node id="ETN2" object_ref="O2"/>
    </entity_relation>
    <entity_relation> <!-- Semantic entity relation -->
        <relation type="SEMANTIC"> Shaking hands with </relation>
20 <entity_node id="ETN3" object_ref="O2"/> <entity_node id="ETN4"
    object_ref="O1"/>
    </entity_relation>
</entity_relation_graph>

```

25 For purposes of efficiency, entity-relation elements 34 may also include one or more other entity-relation elements 34, as depicted in Fig. 2. This allows the creation of efficient nested graphs of entity relationships, such as those utilized in the Synchronized Multimedia Integration Language (SMIL), which synchronizes different media documents by using a series of nested parallel sequential relationships.

30 An object hierarchy 26 is a particular type of entity-relation graph 28 and therefore can be implemented using an entity-relation graph 28, wherein entities are related by containment relationships. Containment relationships are topological

relationships such as those denoted in Table 4. To illustrate that an object hierarchy 26 is a particular type of an entity-relation graph 28, the exemplary object hierarchy 26 depicted in Fig. 1B is expressed below in XML as an entity-relation graph 28.

```
5      <entity_relation_graph>
        <entity_relation>
          <relation type="SPATIAL"> Contain </relation>
          <entity_node object_ref="O0"/> <entity_node object_ref="O1"/>
        </entity_relation>
        <entity_relation>
10      <relation type="SPATIAL"> Contain </relation>
          <entity_node object_ref="O0"/> <entity_node object_ref="O2"/>
        </entity_relation>
      </entity_relation_graph>
```

15 The exemplary hierarchy depicted in Fig. 1B describes how object O0 8 (the overall photograph) spatially contains objects O1 2 ("Person A") and O2 6 ("Person B"). Thus, based on particular requirements, applications may implement hierarchies utilizing either the convenience of the comprehensive structure of an entity-relation graph 28, or alternatively by utilizing the efficiency of object hierarchies 26.

20 For image descriptors associated with any type of features, such as media features 36, visual features 38 or semantic features 40 for example, the image description system of the present invention may also contain links to extraction and similarity matching code in order to facilitate code downloading, as illustrated in the XML example below. These links provide a mechanism for efficient searching and  
25 filtering of image content from different sources using proprietary descriptors. Each image descriptor in the image description system of the present invention may include a descriptor value and a code element, which contain information regarding the extraction and similarity matching code for that particular descriptor. The code elements (<code>) may also include pointers to the executable files (<location>), as  
30 well as the description of the input parameters (<input\_parameters>) and output parameters (<output\_parameters>) for executing the code. Information about the type

of code (namely, extraction code or similarity matching code), the code language (such as Java or C for example), and the code version are defined as particular attributes of the code element.

The exemplary XML implementation set forth below provides a description of a so-called Tamura texture feature, as set forth in H. Tamura, S. Mori, and T. Yamawaki, "Textual Features Corresponding to Visual Perception," IEEE Transactions on Systems, Man and Cybernetics, Vol. 8, No. 6, June 1978, the entire content of which is incorporated herein by reference. The Tamura texture feature provides the specific feature values (namely, coarseness, contrast, and directionality) and also links to external code for feature extraction and similarity matching. In the feature extraction example shown below, additional information about input and output parameters is also provided. Such a description could for example be generated by a search engine in response to a texture query from a meta search engine. The meta search engine could then use the code to extract the same feature descriptor from the results received from other search engines, in order to generate a homogeneous list of results for a user. In other cases, only the extraction and similarity matching code, but not the specific feature values, is included. If necessary in such instances, filtering agents may be used to extract feature values for processing.

The exemplary XML implementation shown below also illustrates the way in which the XML language enables externally defined description schemes for descriptors to be imported and combined into the image description system of the present invention. In the below example, an external descriptor for the Croma Key shape feature is imported into the image description by using XML namespaces. Using this framework, new features, types of features, and image descriptors can be conveniently included in an extensible and modular way.

```
<texture> <tamura>
  <tamura_value coarseness="0.01" contrast="0.39" directionality="0.7"/>
  <code type="EXTRACTION" language="JAVA" version="1.1"> <!-- Link extraction
code -->
  <location> <location_site href="ftp://extract.tamura.java"/> </location>
  <input_parameters> <parameter name="image" type="PPM"/>
</input_parameters>
```

```

        <output_parameters>
            <parameter name="tamura texture" type="double[3]"/>
        </output_parameters>
    </code>
5    <code type="DISTANCE" language="JAVA" version="4.2"> <!-- Link similarity code --
    >
        <location> <location_site href="ftp://distance.tamura.java"/> </location>
    </code>
    </tamura> </texture>

10    <shape> <!-- Import external shape descriptor DTD -->
        <chromaKeyShape xmlns:extShape "http://www.other.ds/chromaKeyShape.dtd">
            <extShape:HueRange>
                <extShape:start> 40 </extShape:start> <extShape:end> 40
            </extShape:end>
15            </extShape:HueRange>
        </chromaKeyShape>
    </shape>

```

The image description system of the present invention also supports  
 modality transcoding. In an exemplary instance in which a content broadcaster must  
 20 transmit image content to a variety of users, the broadcaster must transcode the image  
 content into different media modalities and resolutions, in order to accommodate the  
 users' various terminal requirements and bandwidth limitations. The image  
 description system of the present invention provides modality transcoding in  
 connection with both local and global objects. This modality transcoding transcodes  
 25 the media modality, resolution, and location of transcoded versions of the image  
 objects in question, or alternatively links to external transcoding code. The image  
 descriptor in question also can point to code for transcoding an image object into  
 different modalities and resolutions, in order to satisfy the requirements of different  
 user terminals. The exemplary XML implementation shown below illustrates  
 30 providing an audio transcoded version for an image object.

```

    <image_object type="GLOBAL" id="O0">
        <img_obj_media_features>
            <location> <location_site href="Hi.gif"/> </location>
            <modality_transcoding>

```

```

        <modality_object_set>
            <modality_object id="mo2" type="AUDIO" resolution="1">
                <location><location_site
5      href="Hi.au.xml"?o1/></location>
            </modality_object>
        </modality_object_set>
    </modality_transcoding>
    <img_obj_media_features>
    </image_object>

```

10 Additional exemplary Document Type Definition (DTD) implementations of the image description system of the present invention are provided in Appendix A.

Fig. 5 depicts a block diagram of an exemplary computer system for implementing the image description system of the present invention. The computer system depicted includes a computer processor section 402 which receives digital data representing image content, via image input interface 404 for example. Alternatively, the digital image data can be transferred to the processor section 402 from a remote source via a bidirectional communications input/output (I/O) port 406. The image content can also be transferred to the processor section 402 from non-volatile computer media 408, such as any of the optical data storage or magnetic storage systems well known in the art. The processor section 402 provides data to an image display system 410, which generally includes appropriate interface circuitry and a high resolution monitor, such as a standard SVGA monitor and video card which are commonly employed in conventional personal computer systems and workstations for example. A user input device, such as a keyboard and digital pointing device a mouse, trackball, light pen or touch screen for example), is coupled to the processor section 402 to effect the user's interaction with the computer system. The exemplary computer system of Fig. 5 will also normally include volatile and non volatile computer memory 414, which can be accessed by the processor section 402 during processing operations.

30 Fig. 6 depicts a flow chart diagram which further illustrates the processing operations undertaken by the computer system depicted in Fig. 5 for purposes of implementing the image description system of the present invention. Digital image data 310 is applied to the computer system via link 311. The computer



system, under the control of suitable application software, performs image object extraction in block 320, in which image objects 30 and associated features, such as media features 36, visual features 38 and semantic features 40 for example, are generated. Image object extraction 320 may take the form of a fully automatic processing operation, a semi-automatic processing operation, or a substantially manual operation in which objects are defined primarily through user interaction, such as via user input device 412 for example.

In a preferred embodiment, image object extraction 320 consists of two subsidiary operations, namely image segmentation as depicted by block 325, and feature extraction and annotation as depicted by block 326. For the image segmentation 325 step, any region tracking technique which partitions digital images into regions that share one or more common characteristics may be employed. Likewise, for the feature extraction and annotation step 326, any technique which generates features from segmented regions may be employed. A region-based clustering and searching subsystem is suitable for automated image segmentation and feature extraction. An image object segmentation system is an example of a semi-automated image segmentation and feature extraction system. Manual segmentation and feature extraction could alternatively be employed. In an exemplary system, image segmentation 325 may for example generate image objects 30, and feature extraction and annotation 326 may for example generate the features associated with the image objects 30, such as media features 36, visual features 38 and semantic features 40, for example.

The object extraction processing 320 generates an image object set 24, which contains one or more image objects 30. The image objects 30 of the image object set 24 may then be provided via links 321, 322 and 324 for further processing in the form of object hierarchy construction and extraction processing as depicted in block 330, and/or entity relation graph generation processing as depicted in block 336. Preferably, object hierarchy construction and extraction 330 and entity relation graph generation 336 take place in parallel and via link 327. Alternatively, image objects 30 of the image object set 24 may be directed to bypass object hierarchy construction and extraction 330 and entity relation graph generation 336, via link 323. The object

hierarchy construction and extraction 330 thus generates one or more object hierarchies 26, and the entity relation graph generation 336 thus generates one or more entity relation graphs 28.

The processor section 402 then merges the image object set 24, object hierarchies 26 and entity relation graphs 28 into an image description record for the image content in question. The image description record may then be stored directly in database storage 340, or alternatively may first be subjected to compression by binary encoder 360 via links 342 and 361, or to encoding by description definition language encoding (using XML for example) by XML encoder 350 via links 341 and 351. Once the image description records have been stored in data base storage 340, they remain available in a useful format for access and use by other applications 370, such as search, filter and archiving applications for example, via bidirectional link 371.

Referring to Fig. 7, an exemplary embodiment of a client-server computer system on which the image description system of the present invention can be implemented is provided. The architecture of the system 100 includes a client computer 110 and a server computer 120. The server computer 120 includes a display interface 130, a query dispatcher 140, a performance database 150, query translators 160, 161, 165, target search engines 170, 171, 175, and multimedia content description systems 200, 201, 205, which will be described in further detail below.

While the following disclosure will make reference to this exemplary client-server embodiment, those skilled in the art should understand that the particular system arrangement may be modified within the scope of the invention to include numerous well-known local or distributed architectures. For example, all functionality of the client-server system could be included within a single computer, or a plurality of server computers could be utilized with shared or separated functionality.

Commercially available metasearch engines act as gateways linking users automatically and transparently to multiple text-based search engines. The system of Fig. 7 grows upon the architecture of such metasearch engines and is designed to intelligently select and interface with multiple on-line multimedia search

engines by ranking their performance for different classes of user queries.

Accordingly, the query dispatcher 140, query translators 160, 161, 165, and display interface 130 of commercially available metasearch engines may be employed in the present invention.

5           The dispatcher 140 selects the target search engines to be queried by consulting the performance database 150 upon receiving a user query. This database 150 contains performance scores of past query successes and failures for each supported search option. The query dispatcher only selects search engines 170, 171, 175 that are able to satisfy the user's query, e.g. a query seeking color information will  
10 trigger color enabled search engines. Search engines 170, 171, 175 may for example be arranged in a client-server relationship, such as search engine 170 and associated client 172.

          The query translators 160, 161, 165, translate the user query to suitable scripts conforming to the interfaces of the selected search engines. The display  
15 component 130 uses the performance scores to merge the results from each search engine, and presents them to the user.

          In accordance with the present invention, in order to permit a user to intelligently search the Internet or a regional or local network for visual content, search queries may be made either by descriptions of multimedia content generated by  
20 the present invention, or by example or sketch. Each search engine 170, 171, 175 employs a description scheme, for example the description schemes described below, to describe the contents of multimedia information accessible by the search engine and to implement the search.

          In order to implement a content-based search query for multimedia  
25 information, the dispatcher 140 will match the query description, through the multimedia content description system 200, employed by each search engine 170, 171, 175 to ensure the satisfaction of the user preferences in the query. It will then select the target search engines 170, 171, 175 to be queried by consulting the performance database 150. If for example the user wants to search by color and one  
30 search engine does not support any color descriptors, it will not be useful to query that particular search engine.

Next, the query translators 160, 161, 165 will adapt the query description to descriptions conforming to each selected search engine. This translation will also be based on the description schemes available from each search engine. This task may require executing extraction code for standard descriptors or downloaded  
5 extraction code from specific search engines to transform descriptors. For example, if the user specifies the color feature of an object using a color coherence of 166 bins, the query translator will translate it to the specific color descriptors used by each search engine, e.g. color coherence and color histogram of x bins.

Before displaying the results to the user, the query interface will merge  
10 the results from each search option by translating all the result descriptions into a homogeneous one for comparison and ranking. Again, similarity code for standard descriptors or downloaded similarity code from search engines may need to be executed. User preferences will determine how the results are displayed to the user.

Referring next to Fig. 8, a description system 200 which, in accordance  
15 with the present invention, is employed by each search engine 170, 171, 175 is now described. In the preferred embodiment disclosed herein, XML is used to describe multimedia content.

The description system 200 advantageously includes several multimedia processing, analysis and annotation sub-systems 210, 220, 230, 240, 250,  
20 260, 270, 280 to generate a rich variety of descriptions for a collection of multimedia items 205. Each subsystem is described in turn.

The first subsystem 210 is a region-based clustering and searching system which extracts visual features such as color, texture, motion, shape, and size for automatically segmented regions of a video sequence. The system 210  
25 decomposes video into separate shots by scene change detection, which may be either abrupt or transitional (e.g. dissolve, fade in/out, wipe). For each shot, the system 210 estimates both global motion (i.e. the motion of dominant background) and camera motion, and then segments, detects, and tracks regions across the frames in the shot computing different visual features for each region. For each shot, the description  
30 generated by this system is a set of regions with visual and motion features, and the camera motion. A complete description of the region-based clustering and searching

system 210 is contained in co-pending PCT Application Serial No. PCT/US98/09124, filed May 5, 1998, entitled "An Algorithm and System Architecture for Object-Oriented Content-Based Video Search," the contents of which are incorporated by reference herein.

5           As used herein, a "video clip" shall refer to a sequence of frames of video information having one or more video objects having identifiable attributes, such as, by way of example and not of limitation, a baseball player swinging a bat, a surfboard moving across the ocean, or a horse running across a prairie. A "video object" is a contiguous set of pixels that is homogeneous in one or more features of  
10 interest, e.g., texture, color, motion or shape. Thus, a video object is formed by one or more video regions which exhibit consistency in at least one feature. For example a shot of a person (the person is the "object" here) walking would be segmented into a collection of adjoining regions differing in criteria such as shape, color and texture, but all the regions may exhibit consistency in their motion attribute.

15           The second subsystem 220 is an MPEG domain face detection system, which efficiently and automatically detects faces directly in the MPEG compressed domain. The human face is an important subject in images and video. It is ubiquitous in news, documentaries, movies, etc., providing key information to the viewer for the understanding of the video content. This system provides a set of regions with face  
20 labels. A complete description of the system 220 is contained in PCT Application Serial No. PCT/US 97/20024, filed November 4, 1997, entitled "A Highly Efficient System for Automatic Face Region Detection in MPEG Video," the contents of which are incorporated by reference herein.

25           The third subsystem 230 is a video object segmentation system in which automatic segmentation is integrated with user input to track semantic objects in video sequences. For general video sources, the system allows users to define an approximate object boundary by using a tracing interface. Given the approximate object boundary, the system automatically refines the boundary and tracks the movement of the object in subsequent frames of the video. The system is robust  
30 enough to handle many real-world situations that are difficult to model using existing approaches, including complex objects, fast and intermittent motion, complicated

backgrounds, multiple moving objects and partial occlusion. The description generated by this system is a set of semantic objects with the associated regions and features that can be manually annotated with text. A complete description of the system 230 is contained in U.S. Patent Application Serial No. 09/405,555, filed  
5 September 24, 1998, entitled "An Active System and Algorithm for Semantic Video Object Segmentation," the contents of which are incorporated by reference herein.

The fourth subsystem 240 is a hierarchical video browsing system that parsers compressed MPEG video streams to extract shot boundaries, moving objects, object features, and camera motion, and. It also generates a hierarchical shot-based  
10 browsing interface for intuitive visualization and editing of videos. A complete description of the system 240 is contained in PCT Application Serial No. PCT/US 97/08266, filed May 16, 1997, entitled "A Method and Architecture for Indexing and Editing Compressed Video Over the World Wide Web," the contents of which are incorporated by reference herein.

15 The fifth subsystem 250 is the entry of manual text annotations. It is often desirable to integrate visual features and textual features for scene classification. For images from on-line news sources, e.g. Clarinet, there is often textual information in the form of captions or articles associated with each image. This textual information can be included in the descriptions.

20 The sixth subsystem 260 is a system for high-level semantic classification of images and video shots based on low-level visual features. The core of the system consists of various machine learning techniques such as rule induction, clustering and nearest neighbor classification. The system is being used to classify images and video scenes into high level semantic scene classes such as {nature  
25 landscape}, {city/suburb}, {indoor}, and {outdoor}. The system focuses on machine learning techniques because we have found that the fixed set of rules that might work well with one corpus may not work well with another corpus, even for the same set of semantic scene classes. Since the core of the system is based on machine learning techniques, the system can be adapted to achieve high performance for different  
30 corpora by training the system with examples from each corpus. The description generated by this system is a set of text annotations to indicate the scene class for each

image or each keyframe associated with the shots of a video sequence. A complete description of the system 260 is contained in S. Paek et al., "Integration of Visual and Text based Approaches for the Content Labeling and Classification of Photographs," ACM SIGIR'99 Workshop on Multimedia Indexing and Retrieval. Berkeley, C A (1999), the contents of which are incorporated by reference herein.

The seventh subsystem 270 is a model based image classification system. Many automatic image classification systems are based on a pre-defined set of classes in which class-specific algorithms are used to perform classification. The system 270 allows users to define their own classes and provide examples that are used to automatically learn visual models. The visual models are based on automatically segmented regions, their associated visual features, and their spatial relationships. For example, the user may build a visual model of a portrait in which one person wearing a blue suit is seated on a brown sofa, and a second person is standing to the right of the seated person. The system uses a combination of lazy-learning, decision trees and evolution programs during classification. The description generated by this system is a set of text annotations, i.e. the user defined classes, for each image. A complete description of the system 270 is contained in A. James et al., "Model Based Classification of Visual Information for Content-Based Retrieval," Symp. Elec. Imaging: Multimedia Proc. and App. - Storage Retrieval and for Image and Video Databases VII, IS&T/SPIE '99 (1999), the contents of which are incorporated by reference herein.

Other subsystems 280 may be added to the multimedia content description system 200, such as a subsystems from collaborators used to generate descriptions or parts of descriptions, for example.

In operation, the image and video content 205 may be a database of still images or moving video, a buffer receiving content from a browser interface 206, or a receptacle for live image or video transmission. The subsystems 210, 220, 230, 240, 250, 260, 270, 280 operate on the image and video content 205 to generate descriptions 211, 221, 231, 241, 251, 261, 271, 281 that include low level visual features of automatically segmented regions, user defined semantic objects, high level scene properties, classifications and associated textual information, as described

above. Once all the descriptions for an image or video item are generated and integrated in block 290, the descriptions are then input into a database 295, which the search engine 170 accesses.

It should be noted that certain of the subsystems, i.e., the region-based clustering and searching subsystem 210 and the video object segmentation system 230 may implement the entire description generation process, while the remaining subsystems implement only portions of the process and may be called on by the subsystems 210, 230 during processing. In a similar manner, the subsystems 210 and 230 may be called on by each other for specific tasks in the process.

Although the present invention has been described in connection with specific exemplary embodiments, it should be understood that various changes, substitutions and alterations can be made to the disclosed embodiments without departing from the spirit and scope of the invention as set forth in the appended claims.



**Annex A: Document Type Definition of Image Description Scheme**

image\_ds.dtd:

```
<!-- Image DS -->

<!-- Entities are like macros. They can be referenced by using the notation
5 "%EntityName;" -->
<!-- For clarity, we have chosen not to reference them in this DTD, but they are
referenced in other DTDs derived from this one. -->

<!ENTITY % image_object_elements "
    img_obj_media_features?,
10    img_obj_semantic_features?,
    img_obj_visual_features?">

<!ENTITY % ref_image_object_attributes "
    id ID #IMPLIED
    object_ref IDREF #IMPLIED
15    object_node_ref IDREFS #IMPLIED
    entity_node_ref IDREFS #IMPLIED">

<!ENTITY % img_obj_media_features_elements "
    location?, file_format?, file_size?, resolution?, modality_transcoding?">

<!ENTITY % img_obj_semantic_features_elements "
20    text_annotation?, ?, who?, what_object?, what_action?, where?, why?,
when?">

<!ENTITY % img_obj_visual_features_elements "
```

image\_scl?, color?, texture?, shape?, size?, position?, motion?">

<!-- Image element -->

<!ELEMENT image (image\_object\_set, object\_hierarchy\*, entity\_relation\_graph\*)>

<!ATTLIST image

5 id ID #IMPLIED>

<!-- Image object set element -->

<!ELEMENT image\_object\_set (image\_object+)>

<!ELEMENT image\_object (img\_obj\_media\_features?, img\_obj\_semantic\_features?,  
img\_obj\_visual\_features?">

10 <!ATTLIST image\_object

type (LOCAL|GLOBAL) #REQUIRED

id ID #IMPLIED

object\_ref IDREF #IMPLIED

object\_node\_ref IDREFS #IMPLIED

15 entity\_node\_ref IDREFS #IMPLIED>

<!ELEMENT img\_obj\_media\_features (location?, file\_format?, file\_size?,  
resolution?,

modality\_transcoding?)>

20 <!ELEMENT img\_obj\_semantic\_features (text\_annotation?, who?, what\_object?,  
what\_action?, where?, why?, when?)>

<!ELEMENT img\_obj\_visual\_features (image\_scl?, color?, texture?, shape?, size?, position?,

motion?)>

<!-- Object hierarchy element -->

5 <!-- The attribute type is the hierarchy binding type -->

<!ELEMENT object\_hierarchy (object\_node)>

<!ATTLIST object\_hierarchy

id ID #IMPLIED

type CDATA #IMPLIED>

10 <!ELEMENT object\_node (object\_node\*)>

<!ATTLIST object\_node

id ID #IMPLIED

object\_ref IDREF #REQUIRED>

<!-- Entity relation graph element-->

15 <!-- Possible types of entity relations and entity relation graphs:

- Spatial: topological, directional

- Temporal: topological, directional

- Semantic -->

<!ELEMENT entity\_relation\_graph (entity\_relation+)>

20 <!ATTLIST entity\_relation\_graph

id ID #IMPLIED

type CDATA #IMPLIED>

<!ELEMENT entity\_relation (relation, (entity\_node | entity\_node\_set | entity\_relation)\*)>

```
<!-- entity_relation
      type CDATA #IMPLIED>

<!-- relation (#PCDATA | code)* -->

<!-- entity_node (#PCDATA) -->
5 <!-- entity_node
      id ID #IMPLIED
      object_ref IDREF #REQUIRED>

<!-- entity_node_set (entity_node+) -->

<!-- External object location DTD -->
10 <!-- ENTITY % location SYSTEM "location.dtd" -->
    %location;

<!-- External file feature DTD -->
<!-- ENTITY % file SYSTEM "file.dtd" -->
    %file;

15 <!-- External modality transcoding DTD -->
<!-- ENTITY % modality_transcoding SYSTEM "modality_transcoding.dtd" -->
    %modality_transcoding;

<!-- External object annotation DTD -->
<!-- ENTITY % text_annotation SYSTEM "text_annotation.dtd" -->
20 %text_annotation;
```

```
<!-- External scalable image DTD -->
<!ENTITY % image_scl SYSTEM "image_scl.dtd">
%image_scl;

<!-- External object color DTD -->
5 <!ENTITY % color SYSTEM "color.dtd">
%color;

<!-- External object texture DTD -->
<!ENTITY % texture SYSTEM "texture.dtd">
%texture;

10 <!-- External object shape DTD -->
<!ENTITY % shape SYSTEM "shape.dtd">
%shape;

<!-- External object size DTD -->
<!ENTITY % size SYSTEM "size.dtd">
15 %size;

<!-- External object position DTD -->
<!ENTITY % position SYSTEM "position.dtd">
%position;

<!-- External object motion DTD -->
20 <!ENTITY % motion SYSTEM "motion.dtd">
%motion;
```

```

<!-- External object time DTD -->
<!ENTITY % time SYSTEM "time.dtd">
%time;

<!ENTITY mpeg7 "ISO/IEC JTC1/SC29/WG11 MPEG-7">

5 <!-- Image DS End -->

```

location.dtd:

```

<!-- Description of resources' location-->

<!-- Objects, image, videos can be located/accessed at different locations -->
<!ELEMENT location (location_site*)>

10 <!-- One location site -->
<!ELEMENT location_site EMPTY>
<!ATTLIST location_site
    href CDATA #REQUIRED
    title CDATA #IMPLIED>

15 <!ELEMENT code (location*)>
<!ATTLIST code
    type (EXTRACTION|DISTANCE) "EXTRACTION"
    language (C|JAVA|PERL) #REQUIRED
    version CDATA #REQUIRED>

20 <!-- Description of resources' storage location -->

```

file.dtd:

<!-- Description of resources' file specifications-->

<!-- File format -->

<!ELEMENT file\_format (#PCDATA)>

5 <!-- File size -->

<!ELEMENT file\_size (#PCDATA)>

<!ELEMENT resolution (res\_width, res\_height)>

<!ELEMENT res\_width (#PCDATA)>

<!ELEMENT res\_height (#PCDATA)>

10 <!-- Description of resources' file specifications -->

modality\_transcoding.dtd:

<!-- Description of modality transcoding agent -->

<!-- Representation of image objects in different modalities (audio, text) and  
resolutions -->

15 <!-- We reuse the defined structure entity\_relation\_graph -->

<!ELEMENT modality\_transcoding (modality\_object\_set, entity\_relation\_graph?)>

<!ELEMENT modality\_object\_set (modality\_object\*) >

5

```

<!ELEMENT modality_object (location) >
<!-- ATTLIST modality_object
      id ID #REQUIRED
      type (AUDIO|VIDEO|TEXT|IMAGE|SYNTHETIC_VIDEO) #REQUIRED
      resolution CDATA #IMPLIED
      object_ref IDREF #IMPLIED-->

<!-- Description of semantic transcoding agent -->

```

text\_annotation.dtd:

10

```

<!-- Text annotation features -->

<!-- ELEMENT text_annotation (name | who | where | why | when | what_object |
what_action)*-->

<!-- Name annotation -->
<!-- ELEMENT name (concept*, code*)-->

<!-- People annotation -->
15 <!-- ELEMENT who (concept*, code*)-->

<!-- Location annotation -->
<!-- ELEMENT where (concept*, code*)-->

<!-- Event annotation -->
<!-- ELEMENT why (concept*, code*)-->

```



<!-- Date annotation -->

<!ELEMENT when (concept\*, code\*)>

<!-- Object annotation -->

<!ELEMENT what\_object (concept\*, code\*)>

5 <!-- Action annotation -->

<!ELEMENT what\_action (entity\_relation\*, concept\*, code\*)>

<!-- Concept -->

<!ELEMENT concept (#PCDATA | code)\*>

10 <!ATTLIST concept

language CDATA "english"

generation (automatic|manual) "manual">

<!-- Text annotation features end -->

image\_scl.dtd

<!-- Image scalability features -->

15 <!ELEMENT image\_scl (image\_sclobj, code\*)

<!ELEMENT image\_sclobj (img\_obj\_scltype, img\_obj\_mode, img\_obj\_numlayers,  
coderef, subsamp\_factor, img\_obj\_shape?)>

<!ELEMENT img\_obj\_scltype EMPTY>

<!ATTLIST img\_obj\_scltype

20 typeinfo (DATPARTITION|SPATIAL|SNR) #REQUIRED>

```
<!-- Image scalability (subtype) mode features -->
<!ELEMENT img_obj_mode EMPTY>
<!ATTLIST img_obj_mode
    modeinfo CDATA #REQUIRED>

5  <!ELEMENT numlayers EMPTY>
    <!ATTLIST img_obj_numlayers
        numval #REQUIRED>

    <!ELEMENT codref EMPTY>
    <!ATTLIST codref
10     layernum #REQUIRED>

    <-- subsampling ratio n/m for horizontal and vertical directions -->
    <!ELEMENT subsamp_factor EMPTY>
    <!ATTLIST subsamp_factor
        hor_factor_n CDATA #REQUIRED
        hor_factor_m CDATA #REQUIRED
15     vert_factor_n CDATA #REQUIRED
        vert_factor_m CDATA #REQUIRED>

    <!ELEMENT img_obj_shape(shape)>

    <!-- scalability features end -->
```

20 color.dtd:

```
<!-- Color features -->
```

```
<![ELEMENT color (color_hist | luv_color)*>

<!-- Color histogram feature -->
<![ELEMENT color_hist (color_hist_value, code*)>
<![ATTLIST color_hist
5      length CDATA #REQUIRED
      color_space (RGB|OHTA|HSV|LUV) #REQUIRED
      quantization (uniform|non-uniform) #REQUIRED>

<![ELEMENT color_hist_value (#PCDATA)>
<![ATTLIST color_hist_value
10      format CDATA #REQUIRED>

<!-- LUV color feature -->
<![ELEMENT luv_color (luv_color_value, code*)>

<![ELEMENT luv_color_value (luv_bin*)>
<![ATTLIST luv_color_value
15      length CDATA #IMPLIED>

<![ELEMENT luv_bin ((l, u, v)?)>
<![ATTLIST luv_bin
20      l CDATA #REQUIRED
      u CDATA #REQUIRED
      v CDATA #REQUIRED>

<![ELEMENT l (#PCDATA)>
<![ELEMENT u (#PCDATA)>
```

```
<!ELEMENT v (#PCDATA)>
```

```
<!-- Color features end -->
```

texture.dtd:

```
<!-- Texture features -->
```

5     

```
<!ELEMENT texture (text_tamura*)>
```

```
<!-- Tamura texture feature -->
```

```
<!ELEMENT text_tamura (tamura_value, code*)>
```

```
<!ELEMENT tamura_value (coarseness, contrast, orientation)?>
```

10     

```
<!ATTLIST tamura_value
      coarseness CDATA #REQUIRED
      contrast CDATA #REQUIRED
      orientation CDATA #REQUIRED>
```

```
<!ELEMENT coarseness (#PCDATA)>
```

```
<!ELEMENT contrast (#PCDATA)>
```

15     

```
<!ELEMENT orientation (#PCDATA)>
```

```
<!-- Texture features end -->
```

shape.dtd:

```
<!-- Shape features -->
```

```
<![ELEMENT shape (eigen_analysis | contour | shape_geom | shape_chkey |
shape_polyg | shape_bitmap | shape_fourier)*>

<!-- Eigenvalue analysis shape feature -->
<![ELEMENT eigen_analysis (eigen_analysis_value, code*)>

5 <![ELEMENT eigen_analysis_value (aspect_ratio, circularity, angle)>

<![ELEMENT aspect_ratio (#PCDATA)>
<![ATTLIST aspect_ratio
    value CDATA #REQUIRED>

<![ELEMENT circularity (#PCDATA)>
10 <![ATTLIST circularity
    value CDATA #REQUIRED>

<![ELEMENT angle (#PCDATA)>
<![ATTLIST angle
    value CDATA #REQUIRED>

15 <!-- Outline -->

<![ELEMENT contour (contour_value*, code*)>

<![ELEMENT contour_value (perimeter, outline)>

<![ELEMENT perimeter (#PCDATA)>
<![ATTLIST perimeter
20    value CDATA #REQUIRED>
```

```
<!--ELEMENT outline (point2d*)>
<!--ATTLIST outline
    length CDATA #IMPLIED>

<!--ELEMENT point2d (point_x, point_y)?>
5 <!--ATTLIST point2d
    x CDATA #REQUIRED
    y CDATA #REQUIRED>

<!--ELEMENT point_x (#PCDATA)>
<!--ELEMENT point_y (#PCDATA)>

10 <!-- External geometry shape features -->
<!--ENTITY % shape_geom SYSTEM "shape_geom.dtd">
%shape_geom;

<!-- External shape features DTD based on chromakeyed texture -->
<!--ENTITY % shape_chkey SYSTEM "shape_chkey.dtd">
15 %shape_chkey;

<!-- External polygon shape feature -->
<!--ENTITY % shape_polyg SYSTEM "shape_polyg.dtd">
%shape_polyg;

<!-- External bitmap shape feature -->
20 <!--ENTITY % shape_bitmap SYSTEM "shape_bitmap.dtd">
%shape_bitmap;
```

```
<!-- External fourier shape feature -->
<!ENTITY % shape_fourier SYSTEM "shape_fourier.dtd">
%shape_fourier;

<!-- Shape features end -->
```

## 5 shape\_geom.dtd:

```
<!-- Shape geometry and moment based properties or features -->

<!ELEMENT shape_geom (boundary | sizevalue | symmetry | orientation2d |
orientation3d)*>

<!-- shape geometry based properties: boundary, sizevalue, symmetry -->

10 <!ELEMENT boundary EMPTY>
<!ATTLIST boundary
    perimeter CDATA #REQUIRED>

<!ELEMENT sizevalue EMPTY>
<!ATTLIST sizevalue
15     area CDATA #REQUIRED>

<!ELEMENT symmetry EMPTY>
<!ATTLIST symmetry
    type_fold CDATA #REQUIRED>

<!-- shape moment based properties: orientation2d-->
```

```
<!-- ELEMENT orientation2d EMPTY -->
<!-- ATTLIST orientation2d
      angle CDATA #REQUIRED -->

<!-- orientation 3D features -->
5 <!-- ELEMENT orientation3d (roll, pitch, yaw) -->
  <!-- ELEMENT roll EMPTY -->
    <!-- ATTLIST roll
          value CDATA #REQUIRED
          unit CDATA #FIXED "DEGREES" -->
10 <!-- ELEMENT pitch EMPTY -->
    <!-- ATTLIST pitch
          value CDATA #REQUIRED
          unit CDATA #FIXED "DEGREES" -->
  <!-- ELEMENT yaw EMPTY -->
15 <!-- ATTLIST yaw
          value CDATA #REQUIRED
          unit CDATA #FIXED "DEGREES" -->

  <!-- shape property features end -->
```

shape\_chkey.dtd:

```
20 <!-- Shape based on chromakeyed texture features -->
  <!-- ELEMENT shape_chkey (chromakey) -->

  <!-- chromakey shape features -->
  <!-- ELEMENT chromakey(chromakey_value, code*) -->
```



```
<!-- chromakey_value EMPTY -->
<!-- chromakey_value
    keycolorY CDATA #REQUIRED
    keycolorU CDATA #REQUIRED
5    keycolorV CDATA #REQUIRED
    lowThresh CDATA #REQUIRED
    highThresh CDATA #REQUIRED-->

<!-- shape features end -->
```

shape\_polyg.dtd:

```
10 <!-- Polygon Shape features -->

    <!-- shape_polyg (numvert, vertex*) -->

    <!-- Number of vertices -->
    <!-- numvert EMPTY -->
    <!-- numvert
15     value CDATA #REQUIRED-->

    <!-- coordinates of each vertex -->
    <!-- vertex EMPTY -->
    <!-- vertex
20     xcoord CDATA #REQUIRED
     ycoord CDATA #REQUIRED
     zcoord CDATA #REQUIRED-->
```

```
<!-- Polygon Shape features end -->
```

shape\_bitmap.dtd

```
<!-- Bitmap Shape features -->
```

```
<!ELEMENT shape_bitmap (offset, size, bitmap*)>
```

5 <!-- Offset (top,left) value -->

```
<!ELEMENT offset EMPTY>
```

```
<!ATTLIST offset
```

```
    xoff CDATA #REQUIRED
```

```
    yoff CDATA #REQUIRED>
```

10 <!-- Window size -->

```
<!ELEMENT winsize EMPTY>
```

```
<!ATTLIST winsize
```

```
    xsize CDATA #REQUIRED
```

```
    ysize CDATA #REQUIRED>
```

15 <!-- Bitmap -->

```
<!ELEMENT bitmap EMPTY>
```

```
<!ATTLIST bitmap
```

```
    mask CDATA #REQUIRED>
```

```
<!ELEMENT contourlength EMPTY>
```

20 <!ATTLIST contourlength

```
    length CDATA #REQUIRED>
```

```
<!-- Bitmap Shape features end -->
```

shape\_fourier.dtd:

```
<!-- DFT Shape features -->
```

```
<!ELEMENT shape_fourier (length, quantizer, numcoeff, dft*)>
```

5 <!-- Contourlength -->

```
<!ELEMENT lenght EMPTY>
```

```
<!ATTLIST length
```

```
    value CDATA #REQUIRED>
```

```
<!-- Quantizer stepsize of coefficients -->
```

10 <!ELEMENT quantizer EMPTY>

```
<!ATTLIST quantizer
```

```
    value CDATA #REQUIRED>
```

```
<!-- Number of Coefficients-->
```

```
<!ELEMENT numcoeff EMPTY>
```

15 <!ATTLIST numcoeff

```
    value CDATA #REQUIRED>
```

```
<!-- value of each DFT coefficient -->
```

```
<!ELEMENT dft EMPTY>
```

```
<!ATTLIST dft
```

20 x CDATA #REQUIRED

```
    y CDATA #REQUIRED
```

```
<!-- DFT Shape features end -->
```

size.dtd:

```
<!-- Size features -->

<!-- Size -->
5 <!ELEMENT size (size_dimensions | size_num_pixels)>

<!-- Dimensions (X,Y) -->
<!ELEMENT size_dimensions (size_x, size_y)?>
<!ATTLIST size_dimensions
10     x CDATA #REQUIRED
     y CDATA #REQUIRED>

<!ELEMENT size_x (#PCDATA)>
<!ELEMENT size_y (#PCDATA)>

<!-- Number of pixels -->
<!ELEMENT size_num_pixels (#PCDATA)>
15 <!ATTLIST size_num_pixels
     value CDATA #REQUIRED>

<!-- Size features end -->
```

position.dtd:

```
<!-- Position features -->
```

```
<!ELEMENT position (segmentation_mask_analysis*)>

<!-- Segmentation mask position feature -->

<!ELEMENT segmentation_mask_analysis (segmentation_mask_analysis_value,
code*)>

5 <!ELEMENT segmentation_mask_analysis_value (left-top_vertex, centroid)>

<!ELEMENT left-top_vertex EMPTY>
<!-- ATTLIST left-top_vertex
      x CDATA #REQUIRED
      y CDATA #REQUIRED -->

10 <!ELEMENT centroid EMPTY>
<!-- ATTLIST centroid
      x CDATA #REQUIRED
      y CDATA #REQUIRED -->

<!-- Position features end -->
```

ANNEX B: XML for Example of indexing hierarchy and multi-abstraction levels in  
Figure 3

ex\_index\_hierarchy.xml:

```
<image>
5   <image_object_set>
      <!-- List of objects in the image -->
      <image_object type="LOCAL" id="o1">
          <size> <size_num_pixels> 120 </size_num_pixels> </size>
      </image_object>
10   <image_object type="LOCAL" id="o2">
          <size> <size_num_pixels> 140 </size_num_pixels> </size>
      </image_object>
      <image_object type="LOCAL" id="o3">
          <size> <size_num_pixels> 45 </size_num_pixels> </size>
15   </image_object>
      <image_object type="LOCAL" id="o4">
          <size> <size_num_pixels> 35 </size_num_pixels> </size>
      </image_object>
      <image_object type="LOCAL" id="o5">
20   <size> <size_num_pixels> 9 </size_num_pixels> </size>
      </image_object>
      <!-- Intermediate nodes in the indexing hierarchy -->
      <image_object type="LOCAL" id="o6">
          <size> <size_num_pixels /></size>
25   </image_object>
      <image_object type="LOCAL" id="o7">
```

```

    <size> <size_num_pixels>
        <less_than> 50 </less_than>
    </size_num_pixels> </size>
</image_object>
5  <image_object type="LOCAL" id="o8">
    <size> <size_num_pixels>
        <less_than> 10 </less_than>
    </size_num_pixels> </size>
    </image_object>
10 </image_object_set>
    <!-- Indexing hierarchy based on area of objects (number of pixels) -->
    <!-- The first object of the hierarchy contains the features used in the indexing
-->
    <!-- The other intermediate nodes include constraints on the feature values -->
15 <object_hierarchy>
    <object_node id="on1" object_ref="o6">
        <object_node id="on2" object_ref="o1" />
        <object_node id="on3" object_ref="o2" />
        <object_node id="on4" object_ref="o7">
20     <object_node id="on5" object_ref="o3" />
        <object_node id="on6" object_ref="o4" />
        <object_node id="on7" object_ref="o8">
            <object_node id="on8" object_ref="o5" />
        </object_node>
    </object_node>
25 </object_node>
    </object_node>
    </object_hierarchy>
</image>
```